

字符串: String

JavaScript 参考 - JavaScript | MDN (mozilla.org)

1. 字符串也可以当成数组来访问;
2. split: 使用指定的字符 (就是split的参数) 把字符串分割成数组;
3. toUpperCase: 把字符串转换成大写;
4. toLowerCase: 把字符串转换成小写;
5. includes: 检查是否包含指定的字符串, 有返回true, 没有返回false;
6. indexOf: 返回一个字符串在另外一个字符串中第一次出现的位置; 因为我们知道字符串的下标是从0开始的, 所以如果找不到就返回-1; 如果找到就返回具体的位置;
7. lastIndexOf: 最后一次出现的位置; 找不到还是-1;
8. substring(start,end): 从start截取带end-1;也就是说不包含end指定的位置的那个字符;
9. substr(start, length): 从start位置开始截取length长度的字符串;
10. slice(start,end): 方法提取某个字符串的一部分, 并返回一个新的字符串, 且不会改动原字符串。

1.String的构造方法

- 1) String(String original):把字符串数据封装成字符串对象
- 2) String(char[] value):把字符数组的数据封装成字符串对象
- 3) String(char[] value, int index, int count):把字符数组中的一部分数据封装成字符串对象

2.String类的获取功能:

- 1) length(): 获取字符串的长度, 其实也就是字符个数
- 2) charAt(int index): 获取指定索引处的字符
- 3) indexOf(String str): 获取str在字符串对象中第一次出现的索引
- 4) substring(int start): 从start开始截取字符串
- 5) String substring(int start,int end):从start开始, 到end结束截取字符串。包括start, 不包括end

3.String判断功能

- 1) equals(Object obj):比较字符串的**内容**是否相同
- 2) equalsIgnoreCase(String anotherString):比较字符串的**内容是否相同,忽略大小写**
- 3) startsWith(String prefix):判断字符串对象**是否以指定的字符开头 (区分大小写)**
- 4) startsWith(String prefix,**int toffset**): 判断字符串对象**是否以指定的字符开头**, 参数toffset为指定从**哪个下标开始**
- 5) endsWith(String str):判断字符串对象**是否以指定的字符结尾**
- 6) isEmpty(): 判断指定字符串**是否为空**
- 6) compareTo(String anotherString): 比较字符串的大小, **前者大**返回整数, **后者大**返回负数, 相等返回0

4.String类中的转化方法:

- 1) toCharArray():把字符串转换为**字符数组**
- 2) toLowerCase():把字符串转换为小写字符串
- 3) toUpperCase():把字符串转换为大写字符串

```
1  /*split() 方法用于切分字符串*/
2  var str = new String('width device width initial scale');
3  /* 使用空格（不是空字符）把字符串切割成数组*/
4  console.log(str.split(' '));
5  console.log(str.split());
6  /* 会隐式的字符串转换成字符串对象，也就是String的实例化对象*/
7  var fruit = 'apple,pear,banana,orange,mangguo';
8  console.log(fruit.split(','));
9  console.log(fruit.split());
10
11  /** 把字符串转换成大写*/
12  var fruit = 'apple,pear,banana,orange,mangguo';
13  console.log(fruit.toUpperCase()); //APPLE,PEAR,BANANA,ORANGE,MANGGUO
14
15  var str = 'APPLE';
16  console.log(str.toLowerCase());//apple
```

5.其他常用方法

- 1) trim(): 去除字符串**两端空格**
- 2) split():去除字符串中指定的字符，然后返回一个新的字符串
- 3) subSequence(int beginIndex,int endIndex): 截取字符串中指定位置的字符组成一个新的字符串
- 4) replace(char oldChar, char newChar): 将指定字符替换成另一个指定的字符
- 5) replaceAll(String regex,String replacement): 用新的内容替换全部旧内容
- 6) replaceFirst(String regex,String replacement): 替换首个满足条件的内容
- 7) lastIndexOf(String str): 返回指定字符出现的最后一下的下标
- 8) contains(CharSequence s): 查看字符串中是否都含有指定字符
- 9) concat(String str): 在原有的字符串的基础上**加上指定字符串**

```
1  var str = 'ABCDEFGHJKLMNOPQRSTUVWXYZ';
2
```

```
3 //使用指定的分隔符字符串将一个String对象分割成子字符串数组，以一个指定的分割字符串来决定每个拆分
  的位置
4 var str1 = str.split('-');
5 console.log(str1);
6
7 //从一个字符串中返回指定的字符
8 console.log(str.charAt(1)); // B
9
10 //返回 0 到 65535 之间的整数，表示给定索引处的 UTF-16 代码单元
11 console.log(str.charCodeAt(1)); //66
12
13 //判断当前字符串是否是以另外一个给定的子字符串“结尾”的，根据判断结果返回 true 或 false
14 console.log(str.endsWith('XYZ')); //判断是不是以“XYZ”结尾，true
15
16 //提取某个字符串的一部分，并返回一个新的字符串，且不会改动原字符串
17 console.log(str.slice(20)); //UVWXYZ
18
19 //检查是否包含指定的字符串
20 console.log(str.includes('ABC')); //true
21 console.log(str.includes('KLM')); //true
22 console.log(str.includes('FGH1')); //false
23
24 //从左向右找字符串第一次出现的索引位置
25 console.log(str.indexOf('HI')); //8
26 console.log(str.indexOf('hi', 1)); //从下标1开始找，你就找不到hi，-1
27
28 //从左向右找字符串最后一次出现的索引位置
29 console.log(str.lastIndexOf('a')); //-1 你就找不到a，-1
30
31 //返回一个字符串在开始索引到结束索引之间的一个子集，或从开始索引直到字符串的末尾的一个子集
32 console.log(str.substring(3, 6)); //DEF
33 console.log(str.substring(3)); //DEFGHIJKLMNOPQRSTUVWXYZ
34
35 //返回一个字符串中从指定位置开始到指定字符数的字符
36 console.log(str.substr(3, 3)); //DEF
37
38 /*字符串填充*/
39 //用另一个字符串填充当前字符串(如果需要的话，会重复多次)，以便产生的字符串达到给定的长度。从当前字符串的左侧开始填充。
40 console.log(str.padStart(27, 'a')); //aABCDEFGHIJKLMNOPQRSTUVWXYZ
```

```

41 var str2 = '3';
42 console.log(str2.padStart(2, '0')); //03
43 var str3 = '12';
44 console.log(str3.padStart(2, '0')); //12
45
46 //字符串替换
47 //返回一个由替换值（replacement）替换部分或所有的模式（pattern）匹配项后的新字符串
48 var html = '你好，我是关键词';
49 console.log(html.replace('关键词', '<span style="color:red">关键词</span>'));
50 //你好，我是<span style="color:red">关键词</span>
51
52 //去除字符串前后的空格的
53 var space = '      123 '; //过滤用户输入的信息
54 console.log(space); //      123
55 console.log(space.trim()); //123

```

案例

1. 获取字符串个数

```

1 var str1 = 'nsdisijsonjisoanjnjsnjhnn';
2
3 function strCharCount(str) {
4     var obj = {};
5     for (var i = 0; i < str.length; i++) {
6         var k = str[i];
7         /** 把循环出来的每一个字符作为属性追加到obj对象里面*/
8         /** 如果已经出现过了，
9             就应该在原来的基础之上+1
10            否则，
11            初始化为1
12            */
13         if (obj[k]) {
14             obj[k]++; //把一个变量作为对象的属性时，需要用[]
15         } else {
16             obj[k] = 1;
17         }
18     }
19     return obj;

```

```

20 }
21
22 var result = strCharCount(str1); //函数调用，传递实参给形参，传参
23 console.log(result);
24
25 /**
26 a: 1
27 d: 1
28 h: 1
29 i: 3
30 j: 5
31 n: 7
32 o: 2
33 s: 5
34 */

```

2.字符串里面的英文单词的首字母大写，其他字母全部小写

```

1 var str1 = 'how are yOu? i am Fine. Thank you.';
2
3 function strfirstCharUpper(str) {
4     var r = ' ';
5     // 1，首先把字符串全部转为小写；
6     //2，然后使用空格（不是空字符）把字符串切割为数组；
7     // 3，对数组进行循环，把首字母变成大写
8     var wordLower = str.toLowerCase().split(' '); //先全部转成小写，再用空格分割成数组
9
10    for (var i = 0; i < wordLower.length; i++) {
11        wordLower[i] = wordLower[i][0].toUpperCase() + wordLower[i].substring(1);
12        console.log(wordLower[i]);
13    }
14
15    r = wordLower.join(' '); //用空格把每个英文单词再连接成一个字符串
16
17    return r;
18 }
19
20 var result = strfirstCharUpper(str1);

```

```
21 console.log(result);
22 /*How Are You? I Am Fine. Thank You.*/
23
24 /*
25 How
26 Are
27 You?
28 I
29 Am
30 Fine.
31 Thank
32 You.
33 How Are You? I Am Fine. Thank You.*/
```

1、substr() 方法

语法：substr(start,length)

在字符串中抽取从 start 下标开始的指定数目的字符。

(1) start，提取字符的位置。如果为负值，则被看作 str.Length + start，其中 str.Length 为字符串的长度（例如，如果 start 为 -3，则被看作 str.Length + (-3)）。

(2) length，可选可不选。选择则输入要提取的字符数，不选就是全部。

示例1：

```
1 var str="Hello world!"
2 document.write(str.substr(3))
3 //输出lo world!
4 //此示例，只有前面提取字符的位置（即start），后面的length则没有填，
5 // 所以从第3个开始后面的所有都能截取得到
```

示例2：

```
1 var str = "abcdefghij";
2
3 console.log( str.substr(1,2)); // bc
4 console.log( str.substr(-3,2)); // hi
5 console.log( str.substr(-3)); // hij
6 console.log( str.substr(-20,2)); // ab
7 console.log( str.substr(1)); // bcdefghij
8 console.log( str.substr(20,2)); //
```

2、substring () 方法

语法：str.substring(indexStart[, indexEnd])

(1) indexStart为需要截取的第一个字符的索引，该索引位置的字符作为返回的字符串的首字母

(2) indexEnd，可选。一个 0 到字符串长度之间的整数，以该数字为索引的字符不包含在截取的字符串内。

示例1：

```
1 var str="Hello world!"
2 document.write(str.substring(3))
3 //输出为'lo world!'
4 //因为indexEnd为默认，所以从下标3开始截取，后面的全部截取
```

示例2：

```
1 var str="Hello world!"
2      0123456789
3 document.write(str.substring(3,7))
4 //indexStart为3，所以从三开始取，取到'l'
5 //indexEnd为7，所以取到第六位（注意：是第六位），'w'
6 //所以输出为'lo w'
```

总计：

从以上示例可以总结出，substr与substring的不同之处是：

substr (a, b) ， 可以从下标为a的位置开始截取，一直截取到下标为b的位置，（取头取尾）；

而substring (a, b) ， 是从下标为a的位置开始截取，截取到下标为b-1的位置，（取头取不到尾）。

["1", "2", "3"].map(parseInt) 为何返回[1,NaN,NaN]

[解惑 \["1", "2", "3"\].map\(parseInt\) 为何返回\[1,NaN,NaN\]_清枫草塘的博客-CSDN博客](#)

parseInt() 函数可解析一个字符串，并返回一个整数。

语法

```
parseInt(string, radix)
```

参数	描述
string	必需。要被解析的字符串。
radix	可选。表示要解析的数字的基数。该值介于 2 ~ 36 之间。 如果省略该参数或其值为 0，则数字将以 10 为基础来解析。如果它以 "0x" 或 "0X" 开头，将以 16 为基数。 如果该参数小于 2 或者大于 36，则 parseInt() 将返回 NaN。

返回值

返回解析后的数字。

https://blog.csdn.net/weixin_40688217

js:将数组扁平化并去除其中重复数据，最终得到一个升序且不重复的数组/二维数组去重

set方法

- new一个Set对象，set方法去重，set方法只能去重一维数组
- flat方法扁平化，Infinity规定数组长度的限制
- sort方法正序排列，
- 最后把对象转化成数组：

```
1 var array = [1,25,15,[1,2,15,5],15,25,35,1];
2 var set = new Set(array.flat(Infinity).sort((a,b)=>(a-b)));
3 var arr= [...set];
4
   console.log(arr)// [1, 2, 5, 15, 25, 35]
```

toString()方法：

toString()方法用于将当前对象以字符串的形式返回。该方法属于Object对象，由于所有的对象都继承了Object的对象实例，因此几乎所有的实例对象都可以使用该方法

```
1 // 使用构造函数来创建对象
2 function Person(name, age) {
3     // 设置对象的属性
4     this.name = name;
```



```
5     this.age = age;
6 }
7
8 //创建对象的一个实例对象
9 var p = new Person("张三", 20);
10 console.log(p.toString());
```

```
1 // 字符串
2 var str = "Hello";
3 console.log(str.toString());
4
5 // 数字
6 var num = 15.26540;
7 console.log(num.toString());
8
9 // 布尔
10 var bool = true;
11 console.log(bool.toString());
12
13 // Object
14 var obj = {name: "张三", age: 18};
15 console.log(obj.toString());
16
17 // 数组
18 var array = ["CodePlayer", true, 12, -5];
19 console.log(array.toString());
20
21 // 日期
22 var date = new Date(2013, 7, 18, 23, 11, 59, 230);
23 console.log(date.toString());
24
25 // 错误
26 var error = new Error("自定义错误信息");
27 console.log(error.toString());
28
29 // 函数
30 console.log(Function.toString());
```

Hello

15.2654

true

[object Object]

CodePlayer,true,12,-5

Sun Aug 18 2013 23:11:59 GMT+0800 (中国标准时间)

Error: 自定义错误信息

function Function() { [native code] }